

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL VAN WINKEL

**DESENVOLVIMENTO DE SISTEMA DE INFORMAÇÃO
GEOGRÁFICA PARA EXTRAÇÃO DE ATRIBUTOS REGIONAIS DE
AGRICULTURA E PECUÁRIA**

MARÍLIA
2009

RAFAEL VAN WINKEL

DESENVOLVIMENTO DE SISTEMA DE INFORMAÇÃO GEOGRÁFICA
PARA EXTRAÇÃO DE ATRIBUTOS REGIONAIS DE AGRICULTURA E
PECUÁRIA

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário
Eurípides de Marília, mantido pela
Fundação de Ensino Eurípides
Soares da Rocha, como parte dos
requisitos para obtenção do Título
de Bacharel em Ciência da
Computação.

Orientador:
Ms. Leonardo C. Botega

MARÍLIA
2009



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Rafael Van Winkel

**DESENVOLVIMENTO DE SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA EXTRAÇÃO DE
ATRIBUTOS REGIONAIS DE AGRICULTURA E PECUÁRIA**


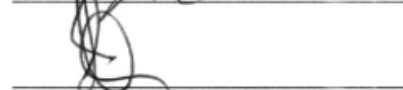

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9,5 (nove e meio)

Orientador: Paulo Augusto Nardi

1º. Examinador: Fabio Lucio Meira

2º. Examinador: Leonardo Castro Botega

Marília, 02 de dezembro de 2009.

"If you enter this world knowing you are loved and you leave this world knowing the same, then everything that happens in between can be dealt with."

Michael Jackson

AGRADECIMENTOS

Agradeço a minha família por me darem a grande oportunidade de ter tido um estudo de qualidade desde meus primeiros passos até aqui.

Aos meus grandes amigos, minha segunda família, por compartilharem comigo os momentos de alegria e desespero durante o desenvolvimento deste projeto.

Ao meu orientador Leonardo pela incrível dedicação à orientação e pela forma com que ele tornou todo este processo mais fácil e agradável.

Aos meus amigos e colegas Mário, Lucas, Ricardo e Marcelo pelo apoio durante estes longos quatro anos.

WINKEL, Rafael van. **Desenvolvimento de Sistema de Informação Geográfica para Extração de Atributos Regionais de Agricultura e Pecuária**. 2009. 52 F. Graduação em Ciência da Computação - Centro Universitário Eurípides de Marília - Fundação de Ensino Eurípides Soares da Rocha, Marília, 2009.

RESUMO

Atualmente, a obtenção de dados agropecuários regionais possui uma série de restrições, no que diz respeito às limitações das técnicas hoje disponíveis. Tendo este problema em vista, o presente trabalho propõe a implementação de um Sistema de Informação Geográfica dedicado à extração de dados regionais de determinadas zonas de influência, escolhidas através de pontos específicos em mapas de relevo. Desta maneira, tal sistema apresentará um ambiente interativo e intuitivo que facilite a extração e interpretação dos dados, buscando contribuir para o desenvolvimento regional através do conhecimento agropecuário das regiões estudadas.

Palavras-chave: Sistema de Informação Geográfica, ASP.NET, Extração de dados Agropecuários.

WINKEL, Rafael van. **Desenvolvimento de Sistema de Informação Geográfica para Extração de Atributos Regionais de Agricultura e Pecuária**. 2009. 52 F. Graduação em Ciência da Computação - Centro Universitário Eurípedes de Marília - Fundação de Ensino Eurípides Soares da Rocha, Marília, 2009.

ABSTRACT

Currently, the regional agriculture data extraction comes with a series of restrictions, based on the limitations we can find on today's available techniques. With that problem in mind, this work proposes a Geographic Information System dedicated to regional data extraction from certain influence zones, selected through specific points on relief maps. Thus, such system will present an efficient and intuitive environment that makes the data extraction and interpretation easier, looking forward to contribute for regional progress through agricultural knowledge of the studied regions.

Palavras-chave: Geographic Information System, ASP.NET, Agricultural Data Retrieval.

LISTA DE ILUSTRAÇÕES

Figura 1. Sistema PIMS-SIG desenvolvido pela ENALTA, que visa uma melhor consulta dos dados através da seleção de pontos específicos do mapa. (Enalta, 2008)	16
Figura 2. Arquitetura .NET (Turtschi <i>et al</i> , 2002).....	28
Figura 3. Trecho de código de um registro Universal Discovery, Description and Integration.	31
Figura 4. Exemplos de códigos em diferentes linguagens utilizando o método Object.ReferenceEquals (MSDN, 2008).	33
Figura 5. Exemplos de documentos com dados agropecuários da região de Marília (LUPA, 2008).....	37
Figura 6. Mapa de Marília e Região utilizado no SIG.	38
Figura 7. Visão geral do funcionamento do SIG proposto.	39
Figura 8. Diagrama de Funcionalidades do Sistema.	40
Figura 9. Tabela de “metadados” do SIG incluindo a divisão das áreas no mapa.	41
Figura 10. Mapa dividido em regiões de acordo com as cidades (EMBRAPA, 2009).....	42
Figura 11. Trecho de código demonstrando a conversão do valor em pixel em coordenada geográfica e a extração de dados a partir do mapa geo-referenciado.	44
Figura 12. Trecho de código demonstrando a conversão da coordenada geográfica em pixel e a extração de dados a partir da coordenada geográfica.	45
Figura 13. Resultado da funcionalidade de busca através do mapa geo-referenciado.	47
Figura 14. Resultado da funcionalidade de busca através de coordenada geográfica.	48
Figura 15. Resultado da busca por atributo “cana-de-açúcar”, passando como limite de valores mil e dez mil hectares.	49

LISTA DE ABREVIATURAS

AAG – Associação Americana de Geógrafos
AML – *Arc Macro Language*
API – *Application Programming Interface*
ASP – *Active Server Pages*
BCL – *Base Class Library*
CAD – *Computer-Aided Design*
CLR – *Common Language Runtime*
COBOL – *Common Business Oriented Language*
ERDAS – *Earth Resources Data Analysis System*
ESRI – *Environmental Systems Research Institute*
GBF-DIME – *Geographic Base File-Dual Independent Map Encoding*
GISCI – *Geographic Information System Certification Institute*
GPS – *Global Positioning System*
HTML – *HyperText Markup Language*
IDE – *Integrated Development Environment*
IN – Inteligência de Negócios
JIT – *Just In Time*
LAN – *Local Area Network*
LIS – *Land Information System*
MSIL – *Microsoft Intermediate Language*
NSGIC – *National States Geographic Information Conference*
SDK – *Software Development Kit*
SIG – Sistema de Informação Geográfica
SQL – *Structured Query Language*
TI – Tecnologia da Informação
TIGER – *Topologically Integrated Geographic Encoding and Referencing*
UCGIS – *University Consortium for Geographic Information Science*
UDDI – *Universal Discovery, Description and Integration*
URISA – *Urban and Regional Information System Association*
USGS – *U.S. Geological Survey*
VB SCRIPT – *Microsoft Visual Basic Scripting Edition*

WWW – *World Wide Web*

XML – *Extensible Markup Language*

SUMÁRIO

INTRODUÇÃO.....	13
CAPÍTULO 1 – SISTEMAS DE INFORMAÇÃO GEOGRÁFICA	14
1.1 Histórico.....	17
1.2. Tipos de Aplicação	22
1.2.1 Cadastral.....	22
1.2.2 Cartografia Automatizada	22
1.2.3 Ambiental.....	23
1.2.4 Redes de Serviço e Utilidades.....	23
1.2.5 Planejamento Rural	24
1.2.6 Business Geographic	25
CAPÍTULO 2 – DOT NET	26
2.1 A Plataforma .NET	26
2.2 Desenvolvimento Web.....	29
2.3 O ambiente ASP.NET.....	30
2.4 C, C++ e C#	31
2.5 A IDE Visual Studio.NET	34
CAPÍTULO 3 – DADOS REGIONAIS	35
3.1 Modelagem de Dados	35
3.2 Disponibilidade de dados agropecuários em Marília e região	36
CAPÍTULO 4 – METODOLOGIA	38
4.1 Introdução	38
4.2 Funcionalidades	40
4.2.1 Extração de dados a partir de mapa geo-referenciado.....	40
4.2.2 Extração de dados a partir de coordenada geográfica.....	42
4.2.3 Extração de dados a partir de atributos geográficos e agropecuários	43

CAPÍTULO 5 – RESULTADOS	46
5.1 Extração de dados a partir de atributos agropecuários.....	46
5.2 Extração de dados a partir de coordenada geográfica.....	46
5.3 Extração de dados a partir de atributos geográficos e agropecuários	46
CAPÍTULO 6 – CONCLUSÕES	50
REFERÊNCIAS BIBLIOGRÁFICAS	51

INTRODUÇÃO

Atualmente, as formas com que os dados agropecuários são disponíveis para a população são pouco intuitivas e de difícil associação e interpretação. As informações estão disponíveis através de documentos e tabelas, divididos por localização geográfica ou atributos geográficos, que podem trazer complicações quando se necessário interpretá-los, especialmente quando houver um grande número de dados.

Devido a estes problemas, existe a necessidade da criação de um ambiente mais intuitivo e dinâmico para extrair estas informações. Além disso, este novo sistema é capaz de exibir as informações de maneira mais rápida e com maior confiabilidade.

Este trabalho tem como objetivo a criação de um Sistema de Informação Geográfica (SIG) voltado a WEB para resolver os problemas descritos acima e colaborar com o desenvolvimento regional através da extração de dados agropecuários de uma maneira mais simplificada do que a existente hoje.

No capítulo 1 é explicado a definição de um SIG, como ele funciona e sua importância no avanço tecnológico. Além disso, este capítulo inclui diferentes tipos de aplicação para estes sistemas e seu histórico, ou seja, como surgiram e sua evolução até o presente momento.

No capítulo 2 são citadas e explicadas as tecnologias utilizadas na implementação do sistema e as vantagens destas escolhas. Este capítulo inclui explicações sobre a linguagem ASP.NET e a IDE Microsoft Visual Studio.

No capítulo 3 está descrita a importância dos SIGs na organização dos dados e como ele se comporta em relação aos mesmos. Além disso, é levantado o atual problema na extração dos dados em Marília e região e os motivos pelo qual um SIG seria mais apropriado para esta tarefa.

No capítulo 4 é mostrado como este sistema foi implementado e suas funcionalidades. Também neste capítulo estão inclusos trechos de códigos que tornaram possível o georreferenciamento das imagens.

No capítulo 5 são mostrados os resultados da implementação do SIG, com imagens de exemplo para demonstrar o comportamento do sistema em cada tipo de extração de dados disponível.

No capítulo 6 são apresentadas as conclusões deste projeto e como ele pode ser útil na extração de dados agropecuários e auxiliar no desenvolvimento regional.

CAPÍTULO 1 – SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

Um Sistema de Informação Geográfica (SIG) é um sistema computacional capaz de visualizar dados referentes a uma determinada região, e é organizado em cinco componentes: interface com o usuário, entrada e integração de dados, consulta e análise espacial, visualização e plotagem, e armazenamento e recuperação de dados. Apesar de estes componentes serem implementados de diferentes maneiras, seguindo as preferências e opções de cada programador, todos eles devem fazer parte do sistema. Na interface com o usuário, é definido como o sistema será operado e configurado. No nível mais intermediário, ou seja, onde é importante o controle e processamento dos dados, o SIG deve ser capaz de editar, analisar, visualizar, armazenar e buscar todas estas informações. Por fim, na camada mais interna do sistema, o SIG deve contar com um sistema de banco de dados geográfico para o armazenamento e a recuperação dos dados e atributos geográficos (Câmara *et al*, 1999).

O SIG trabalha com o usuário através de funções de consulta e seleção restringindo certa área, para consultas através de localizações geográficas, ou através de um intervalo pertencente ao conjunto de dados, levando em conta, neste caso, o dado em si, e não sua localização geográfica. Podemos citar alguns exemplos, como uma consulta sobre a criação de gado na cidade de Marília (restrição por espaço geográfico), ou uma consulta que retorna quais cidades da região possuem mais de 100.000 cabeças de gado (restrição por dados numéricos).

Com o grande avanço da tecnologia no decorrer dos anos e a utilização de novos sistemas e ferramentas, a necessidade de uma rápida obtenção, compreensão, interpretação e visualização de dados aumentaram exponencialmente. Paralelamente, tais avanços tecnológicos incluem Sistema de Posicionamento Global (GPS), radares, sensoriamento remoto, entre outros. O uso de Sistemas de Informações Geográficas (SIG) está se tornando indispensável para vários negócios em várias áreas. Primeiramente utilizados na elaboração de mapas, hoje os SIGs possuem inúmeras vantagens e utilizações em várias áreas como agricultura, cartografia, geologia, logística, planejamentos, meio ambiente, entre muitas outras. (Pitz e Figueiredo, 2001)

A principal característica de um Sistema de Informação Geográfica é que seus dados são georreferenciados, ou seja, são associados e representados a partir de uma localização específica em uma projeção cartográfica ou imagem proveniente de sensoriamento remoto.

Desta maneira, é possível realizar uma série de funções e combinações com estes dados geográficos como consultas, armazenamentos, análises, entre outras. O desenvolvimento e aplicação de um SIG não envolvem apenas a computação e geografia, mas sim uma série de outras tecnologias, tais como o sensoriamento remoto, estatística, fotogrametria, pesquisa operacional, cartografia, inteligência artificial, entre outras.

O grande desafio de um Sistema de Informações Geográficas é a integração de múltiplos tipos de dados, provenientes de fontes e bancos de dados diferentes, em um único sistema eficiente. O SIG deve ser capaz de interpretar todos estes dados e projetar uma visualização fácil e prática de acordo com a área escolhida pelo usuário, seja ela através de seus atributos ou localização geográfica (Câmara *et al*, 1999).

É possível analisar inúmeras vantagens que esta interpretação de diversas informações pode trazer. Ela é importante não só pela melhor visualização dos dados em si, mas também pelas informações que podem ser geradas a partir desta combinação. O SIG pode fornecer parâmetros para inúmeros cálculos de forma rápida, como por exemplo, para geração de relatórios e avaliações em regiões afetadas por alguma praga ou doença, facilitando assim o seu controle e erradicação.

Além do exemplo mostrado anteriormente, podemos citar também seu uso no planejamento de terras, como na alocação de áreas de reserva ambiental, áreas próprias para o plantio, possíveis áreas para a instalação de indústrias, entre outras. Este tipo de decisão envolve uma série de atributos e cálculos que seria ineficiente e inviável se feita por humanos. Com o auxílio dos Sistemas de Informação Geográfica podemos chegar a um resultado mais confiável e em um menor prazo de tempo, o que na maioria das vezes é essencial. (Drummond e French, 2008)

Na agricultura, os SIGs não só trouxeram grandes avanços, mas também acabou contribuindo para o desenvolvimento da agricultura de precisão. O principal conceito desta nova área é a automatização das formas de coleta de dados através do sensoriamento remoto, sistema de posicionamento global (GPS) e o geoprocessamento. (Tschiedel e Ferreira, 2002) Com essas tecnologias podemos garantir maior confiabilidade nas informações e um melhor desempenho na tomada de decisões.

O sistema PIMS-SIG da Enalta (Figura 1), por exemplo, engloba as principais funções para consulta, inclusive através de dados temporais. Deste modo é possível o levantamento de dados de meses ou até anos atrás e a comparação com o desenvolvimento atual, o que pode ser um parâmetro muito importante para a tomada de decisões em um sistema voltado à agricultura de precisão (Enalta, 2008).

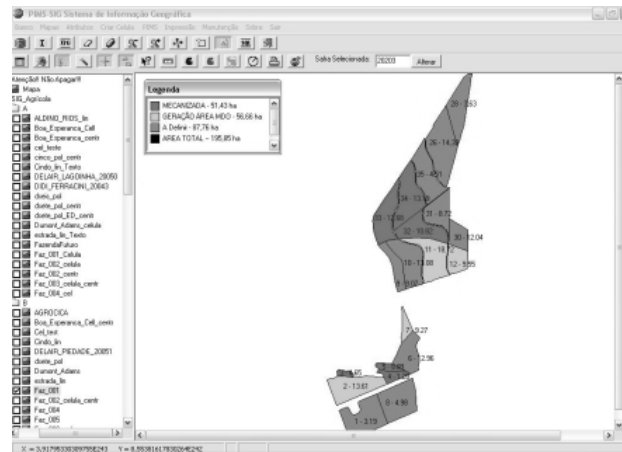


Figura 1. Sistema PIMS-SIG desenvolvido pela ENALTA, que visa uma melhor consulta dos dados através da seleção de pontos específicos do mapa. (Enalta, 2008)

Paralelamente, é possível observar a ausência de ferramentas especializadas, provenientes das mais diversas áreas do conhecimento, dedicadas ao estudo e observação de conjuntos de dados regionais, tais como a agricultura e pecuária, comércio, indústria e demografia.

Nota-se que tais conjuntos de dados, em sua maior parte, são confeccionados e disponibilizados para o conhecimento na forma impressa. Tal documento classifica as diversas perspectivas regionais sob a forma de gráficos e tabelas ilustrativas, processo pouco intuitivo e atrativo, o que acaba não contribuindo com o principal intuito desses dados, a disseminação instrutiva de informações locais.

Atualmente, a obtenção de dados regionais pode contribuir para o desenvolvimento regional sob as mais diversas perspectivas, uma vez que o conhecimento das características do local, como relevo, solo e vegetação, podem auxiliar um produtor em potencial a classificar as determinadas regiões de interesse a fim de tomar a decisão sobre qual local instalar uma benfeitoria e/ou rebanho.

Desta maneira, o presente trabalho de conclusão de curso tem por objetivo o desenvolvimento de um sistema de informação geográfica (SIG) para a visualização bidimensional de dados geográficos e agropecuários relativos a regiões pré-estabelecidas. Armazenadas em um banco de dados, estas informações serão visualizadas pelo usuário a partir da região ou ponto específico determinado por ele, facilitando assim a interpretação dos mesmos.

1.1 Histórico

Os SIGs foram originalmente desenvolvidos como uma tecnologia ambiental. Roger Tomlinson criou o termo *sistemas de informação geográfica* no começo da década de 60, durante um projeto para mapear os recursos naturais do Canadá. Nesta mesma década, Edgar Horwood, um professor de engenharia civil e planejamento da Universidade de Washington, escreveu um dos mais antigos softwares de mapeamento, fundou a *Urban and Regional Information Systems Association*, e conduziu inúmeros mini-cursos e conferências. (Chrisman, 1998) (Tomlinson, 1967) (Chrisman, 2006).

A criação de SIGs modernos coincidiu com a revolução ambiental no começo da década de 70, e os primeiros tipos de aplicação foi para adequação do uso de solo. Técnicas manuais para a análise de adequação de solo são típicas do século 19 (Carr e Zwick, 2007). Entretanto, McHarg (1969) popularizou significativamente esta técnica em seu livro “*Design with Nature*”, fazendo com que as pessoas convertessem o método físico pelo método baseado em computação, começando assim o uso dos SIGs. As primeiras aplicações deste método em problemas de planejamento foram demonstradas por Ingmire e Patri (1971) e Lyle e von Wodtke (1974).

A análise de adequação do uso do solo é uma instância específica dos problemas gerais da integração de dados com localizações. Durante a próxima década, pesquisadores do *Harvard Laboratory for Computer Graphics and Spatial Analysis* desenvolveram duas soluções diferentes para resolver este problema. A implementação a partir da rasterização divide certa área de estudo em uma matriz de células regulares e armazena informações para cada célula do *grid*. Programar uma estrutura de dado utilizando a estrutura de rasterização era relativamente simples, e o desenvolvimento de técnicas de *overlay* e *buffer* não eram complicadas. A rasterização também era a estrutura nativa de dados gerada pela nova *Landsat* satélites.

A segunda e competitiva solução era a utilização de vetores. Mapas vetoriais armazenavam informações geoespaciais como pontos individuais, linhas e objetos poligonais, ligados a um banco de dados de atributos ao invés de conter informações sobre cada célula em um *grid*. Este método era muito mais difícil de programar, e técnicas como *overlay* e análises de *buffer* eram mais complexas e difíceis de serem executadas em termos de tempo de máquina. No entanto, essas dificuldades foram eventualmente superadas com o uso de estruturas de dados topológicas (Chrisman, 2006). Do mesmo modo que hoje os usuários de internet geram mapas que ligam uma cidade a outra através de fotos aéreas, eles são

beneficiados por décadas de desenvolvimento de estruturas de dados criadas durante a construção dos SIGs.

As técnicas de rasterização e vetorização foram desenvolvidas em Harvard, e então comercializadas por arquitetos que tinham a intenção de formar empresas de softwares de sucesso. As primeiras aplicações de SIG estão ligadas com o nome de muitas destas empresas, como a ESRI (*Environmental Systems Research Institute*) e a ERDAS (*Earth Resources Data Analysis System*). No entanto, o uso do SIG como uma tecnologia voltada a área comercial era limitada.

A evolução dos Sistemas de Informação Geográfica é normalmente dividida em três gerações. A primeira geração, iniciada na década de 80, foi caracterizada pelos sistemas referentes à cartografia automatizada. O foco principal até então era mais voltado à produção de mapeamento básico, permitindo depois a entrada de dados, o que se assemelha muito aos sistemas do tipo CAD (*Computer-Aided Design*). Nestes sistemas, porém, o suporte de banco de dados é limitado. Nesta época não havia muito a necessidade da criação de arquivos digitais de dados, ou seja, estas informações eram usadas principalmente em projetos isolados. Inicialmente criado para a plataforma VAX, passou a ser compatível com a plataforma PC/DOS a partir de 1985.

Os SIGs não podiam ser vendidos para arquitetos, que achavam os sistemas CAD mais aptos a suas necessidades de design. A tecnologia SIG se estabilizou em várias organizações de recursos naturais, particularmente na USGS (*U.S. Geological Survey*) e no Serviço Florestal dos Estados Unidos, como também em uma série de empresas privadas madeireiras. Mas o primeiro mercado em grande escala dos softwares SIG veio em 1980 pelos 3000 condados e 19000 municípios do país, em particular os assessores de impostos, planejadores municipais, e engenheiros associados ao governo. Essa mesma área de interesse entre os SIGs e as necessidades dos planejadores locais foi excelente.

Planejadores tinham interesse em questões ambientais, grande compromisso com áreas geográficas específicas, e acesso a ferramentas de computação governamentais. Eles também entendiam sobre mapas e tinham uma grande necessidade de conduzir complexas análises geoespaciais que eram além das capacidades de simples programas de mapeamento da época (Dueker, 1987). Começando pela edição de 1979, o manual padrão para uso em terrenos incluía as análises de adequação do uso do solo como um método fundamental de planejamento de terras, essencialmente substituindo a recomendação anterior para conduzir uma classificação do terreno (Chapin, 1965) (Chapin e Kaiser, 1979).

Um significativo número de sistemas SIGs também foram vendidos para agências do governo estadual e federal, mas estes mercados foram mais limitados do que os para governos locais. No começo dos anos 60, a *U.S. Census Bureau* desenvolveu o arquivo *Geographic Base File-Dual Independent Map Encoding* (GBF-DIME), que foi intencionalmente criado para auxiliar o censo de 1980. Ele era composto por segmentos de rua (e outras estruturas lineares) que definiam blocos de censo e todas as demais áreas para a coleta das estatísticas. Cada segmento linear incluía faixas de endereço, permitindo respostas do censo baseadas no endereço serem alocadas em seus blocos apropriados. Nos censos de 1990 e subsequentes, o arquivo DIME foi substituído pelo arquivo *Topologically Integrated Geographic Encoding and Referencing* (TIGER). Hoje em dia, esta simples estrutura de dados topológica marca o endereço da busca, roteando e as direcionando às aplicações disponíveis na internet, dispositivos GPS e celulares.

Na segunda geração, os Sistemas de Informação Geográfica chegaram de vez ao mercado, inicialmente caracterizados pelo suporte aos ambientes cliente-servidor, podendo desta maneira haver uma interação entre as informações de diferentes mapas. Além de incluírem alguns pacotes adicionais, como por exemplo, ferramentas para processamento de imagens provenientes de sensoriamento remoto, os SIGs a partir de então eram acoplados a gerenciadores de bancos de dados relacionais para o armazenamento de informações sobre os objetos geográficos. Ainda neste aspecto surgiu a possibilidade de consulta (SQL), e um considerável avanço na manipulação e visualização de objetos espaciais. A integração dos dados com imagens evoluiu significativamente nesta geração. Entre outros avanços, os SIGs a partir de então foram criados para ambientes multi-plataformas, destacando as interfaces com janelas.

No começo da década de 90, os SIGs começaram a expandir no mercado dos negócios e, assim que ficaram disponíveis em computadores pessoais, se tornaram viáveis a um conjunto bem maior de usuários corporativos (Castle, 1993). Indústrias com grande poder econômico e claras necessidades geoespaciais, como por exemplo empresas de utilidades públicas, empresas de transporte, e firmas de logística, foram as primeiras a adotarem os sistemas.

Os softwares SIG foram originalmente desenvolvidos como uma aplicação proprietária e especializada, com sua própria linguagem de programação arcaica (ex: linguagem macro Arc/Info, AML), que a isolou da tendência da tecnologia da informação (TI). Em vários governos e universidades locais, departamentos centrais de TI focavam em tendências de sistemas gerenciadores de banco de dados e aplicações de automação (ex:

processadores de texto e planilhas), deixando os que precisavam de apoio na área geoespacial que se virassem com suas próprias aplicações. Isso limitou severamente o potencial de programadores e desenvolvedores de SIGs.

No entanto, nos últimos cinco anos, o software SIG comercial foi se encaixando nas tendências das plataformas de desenvolvimento de software. A velocidade, desempenho, e os benefícios de equipe da integração dos SIG com as tendências da tecnologia da informação permitiram que grandes organizações armazenassem seus dados em sistemas de banco de dados relacionais, como Oracle e SQL Server. Isso também facilitou a integração dos SIGs com grandes bancos de dados variados utilizados pela maioria das agências e firmas. (Drummond e French, 2008)

Nos anos 90, SIGs eram mais usados como aplicações autônomas em computadores (PC) ou estações de trabalho, mas agora o SIG mudou para a arquitetura servidor-cliente. Nessa configuração, os programas que oferecem as funcionalidades de um SIG residem em um servidor central e os usuários acessam este servidor através de máquinas clientes que enviam seus pedidos de processamento e recebem os resultados para visualização. A recente arquitetura de serviços web é ainda mais descentralizada. Nesta tecnologia, funções de software de análise espacial residem em máquinas que estão distribuídas pela web. Protocolos comuns para troca de informações são necessários para este modelo para alcançar um potencial completo.

A nova arquitetura de softwares distribuídos está trazendo implicações em termos de equipe que está afetando profundamente o relacionamento entre os profissionais de planejamento e a tecnologia SIG. O modelo cliente-servidor depende de grandes profissionais de TI que trabalham diretamente com o software SIG e dados em servidores centralizados. Estes profissionais criam programas de aplicação que permitem usuários acessar e manipular o servidor de dados geoespaciais.

Estes usuários, por sua vez, devem possuir habilidades avançadas em gerenciamento de banco de dados para configurar, operar e manter o sistema cliente, e habilidades de programação para criar aplicações que dão suporte às necessidades do usuário final. Normalmente não é assim que acontece. Eles usam os programas de aplicação desenvolvidos por profissionais de TI para criar funções de análises básicas, mas possuem pequenas influências no design dos principais bancos de dados, e possuem talvez um conhecimento rudimentar em SIG.

Já na terceira geração podemos destacar o funcionamento dos SIGs em ambientes remotos, redes locais, e WWW (*World Wide Web*). Além da necessidade de bancos de dados

com mais funções e melhores desenvolvidos, estes sistemas devem ser capazes de compartilhar e receber informações de outros SIGs, fazendo assim uma melhor organização do processamento e um melhor uso dos dados. Está mais do que claro hoje em dia que, devido a inúmeros fatos ambientais e ecológicos, conseguir certos tipos de informação não dependem apenas de processamentos simples e um conjunto de dados relativamente pequeno. Hoje existe a necessidade de entender o que acontece e de algum modo tentar prever numa mesma proporção como será o cenário em determinado período de tempo. Esses detalhes enfatizam a necessidade de um banco de dados e um processamento mais rápido.

Conforme a tecnologia SIG fica mais complexa, estamos vendo uma grande necessidade de uma nova classe de profissionais: os profissionais de SIG. Um profissional de SIG foca na tecnologia e seus usos além de uma aplicação particular dessa tecnologia, como planejamento municipal, por exemplo. Em 2002 a Associação Urbana e Regional dos Sistemas de Informação (URISA) criou o Instituto de Certificação SIG (GISCI), com requisitos educacionais que eram bastante vistos no modelo curricular da *University Consortium for Geographic Information Science (UCGIS)*. Desde então, a Associação Americana de Geógrafos (AAG), *National States Geographic Information Conference (NSGIC)*, e UCGIS se juntaram a instituição de certificação.

Em outra grande mudança drástica que ocorreu nos últimos anos, os SIGs mudaram diretamente para o mercado consumidor. Aparelhos GPS para carros foram os presentes mais populares no Natal de 2007. Estes sistemas calculam a localização do usuário utilizando os bancos de dado SIG que incluem uma rede bem detalhada de ruas, as complexas estruturas geométricas de conexões de ruas e estradas, e informações de locais como posto de gasolina, restaurantes, bancos, lojas e maiores pontos de interesse.

Em Fevereiro de 2008 existiam 1.920 profissionais de SIG certificados e praticando nos Estados Unidos de acordo com a GISCI. Muitas grandes empresas que operam SIGs agora empregam especialistas que combinam conhecimento em uma área particular, como planejamento ou tributação, com especialistas em SIG. Nós acreditamos que organizações que possuem estes especialistas em sua equipe, que entendem dados SIG e análises geoespaciais também em determinado campo como planejamento, estão bem equipadas para implementar SIGs com sucesso.

1.2. Tipos de Aplicação

As aplicações de ciências geoespaciais passaram por um avanço rápido e significativo nas últimas décadas. Podemos separar, de uma forma geral, estas áreas de aplicação em (Câmara *et al*, 1999):

1.2.1 Cadastral

Aplicações de cadastro urbano e rural, realizadas tipicamente por Prefeituras e órgãos governamentais, em escalas que usualmente variam de 1:1.000 a 1:20.000. A capacidade básica de SIG's para atender este setor é dispor de funções de consulta a bancos de dados espaciais e apresentação de mapas e imagens digitais. Nas últimas décadas surgiu uma grande necessidade da criação de cadastramentos de mapas, o que não só auxilia os Sistemas de Informações de Terra (*Land Information System, LIS*), como também são essenciais para o seu desenvolvimento. O grande problema nesta área é que os mapas foram feitos em papel, o que pode causar deterioração com o tempo, e as informações contidas neles podem não ser fielmente verdadeiras por uma série de razões. Os Sistemas de Informação Geográfica auxiliam muito nessa área, podendo armazenar informações mais fiéis com mais confiabilidade, disponibilizar os mapas em diferentes escalas, segurança das informações, entre outras. O único impasse dessa modernização é a disponibilidade de certos mapas em qualidade e escala suficiente para o processo de mapeamento computadorizado (Câmara *et al*, 1999).

1.2.2 Cartografia Automatizada

Realizada por instituições produtoras de mapeamento básico e temático. Neste caso, é essencial dispor de ferramentas de aerofotogrametria digital e técnicas sofisticadas de entrada de dados (como digitalizadores ópticos) e de produção de mapas (como gravadores de filme de alta resolução). Existem dois tipos básicos da utilização de SIGs na cartografia automatizada: a automatização do processo de criar mapas, e a atualização de mapas devido a inúmeras necessidades, através de informações provenientes de resultados de análises e manipulação de dados. No começo da década de 60 surgiram os primeiros esforços para a criação de um processo que automatizasse a criação de mapas. Este processo automatizado permite a manipulação de objetos dentro de um mapa digital sem a necessidade de reformulá-lo, e também traz certa flexibilidade quando necessário alterar sua escala e modo de projeção,

tornando estes processos que antigamente era impossível sem a necessidade da criação de um novo mapa, em funções simples (Câmara *et al*, 1999).

1.2.3 Ambiental

Instituições ligadas às áreas de Meio-Ambiente, Ecologia e Planejamento Regional, que lidam com escalas típicas de 1:10.000 a 1:500.000. As capacidades básicas do SIGs para atender a este segmento são: integração de dados, gerenciamento e conversão entre projeções cartográficas, modelagem numérica de terreno, processamento de imagens e geração de cartas. Esse simples tema abre um enorme leque de áreas onde o SIG ajuda no gerenciamento das informações necessárias, como conservação e monitoramento, gerenciamento de áreas reservadas a vida selvagem, gerenciamento de florestas, controle de poluição da água e do ar, mudanças climáticas, entre outras. Antes dos SIGs, o processo de implantação de indústrias ou qualquer outro processo onde era necessário uma pesquisa e análise para verificar se o meio ambiente seria prejudicado, era feito de forma pouco confiável e na maioria dos casos gerava uma tomada de decisão suspeita. Com os SIGs foi possível eliminar esses problemas e fornecer uma melhor determinação de áreas industriais e áreas de reserva ambiental. Além disso, as informações ficaram mais confiáveis e mais fáceis de serem armazenadas e visualizadas, favorecendo assim uma tomada de decisão mais fiel à realidade (Câmara *et al*, 1999).

1.2.4 Redes de Serviço e Utilidades

Neste segmento, temos as concessionárias de serviços (Água, Energia Elétrica, Telefonia, Transporte). As escalas de trabalho típicas variam entre 1:1.000 a 1:5.000. Cada aplicação de rede tem características individuais e com alta dependência de cada usuário. Os SIGs desenvolvidos para estas redes devem apresentar uma forte ligação com bancos de dados relacionais e a capacidade de adaptação e personalização. O pacote básico disponível com os SIGs deste segmento é insuficiente para a realização da maioria das aplicações, necessitando assim que seus usuários realizem desenvolvimentos significativos em suas linguagens para tornar a aplicação eficiente para sua rede específica. Em redes de energia elétrica, por exemplo, a utilização dos SIGs trazem um benefício enorme na busca de informações através das aplicações com mapas. Por exemplo, se um engenheiro precisa saber a data de instalação de um determinado transformador, tudo o que ele tem a fazer é clicar no ícone deste

transformador na aplicação e os dados atachados a este equipamento serão exibidos na tela. Ele pode também procurar quais foram os outros 100 transformadores instalados na região nesta mesma data. Esta simples busca irá mostrar apenas os transformadores que lhe interessa, escondendo os outros que não atendem a sua necessidade no momento. Em uma única busca ele pôde identificar a localização e o atributo desejado em poucos segundos (Câmara *et al*, 1999).

1.2.5 Planejamento Rural

Neste segmento, temos as empresas agropecuárias que necessitam planejar a produção e distribuição de seus produtos. As escalas de trabalho típicas variam entre 1:1.000 a 1:50.000. Cada aplicação possui características individuais e com alta dependência de cada usuário. Os SIGs desenvolvidos para esta finalidade devem apresentar uma forte ligação com bancos de dados relacionais e a capacidade de adaptação e personalização. O pacote básico disponível com os SIGs deste segmento é insuficiente para a realização da maioria das aplicações, necessitando assim que seus usuários realizem desenvolvimentos significativos em suas linguagens para tornar a aplicação eficiente para seu uso específico. Um dos grandes problemas hoje em dia envolvendo a agricultura é a irrigação. Com o grande crescimento populacional e o desenvolvimento das indústrias, o consumo de água cresce constantemente. Uma grande parte da água própria para o consumo é utilizada na agricultura, o que requer uma forma mais eficiente de conduzi-la para o consumo e para a irrigação. Normalmente, a eficiência da condução de água para a irrigação é muito baixa, o que significa que grande parte da água é desperdiçada antes de chegar ao usuário. Existem modelos de criação de um SIG para o gerenciamento das redes de irrigação. Com este sistema, a eficiência da condução de água em cada segmento é calculada, que por sua vez calcula a eficiência da condução de cada canal, disponibilizando finalmente a eficiência total da rede. Tendo a quantidade de água necessária de cada produtor, a quantidade de água necessária na rede é computada. Então, é possível comparar a quantidade total de água que precisa estar disponível com o total alcançável do açude. Estes dados, por sua vez, podem ser utilizados para planejar e racionar o consumo de água pelos usuários e classificar os produtores de acordo com suas prioridades de irrigação em uma rede (Câmara *et al*, 1999).

1.2.6 Business Geographic

Neste segmento, temos as empresas que necessitam distribuir equipes de vendas e promoção ou localizar novos nichos de mercado. As escalas de trabalho típicas variam entre 1:1.000 a 1:10.000. Cada aplicação também possui características próprias e uma alta dependência de cada usuário. As ferramentas de SIG devem prover meios de apresentação dos bancos de dados espaciais para fins de planejamento de negócios. Em especial, os SIGs devem ser adaptados ao cliente, com ferramentas de particionamento e segmentação do espaço para a localização de novos negócios e alocação de equipes. A necessidade de processos colaborativos e coordenados de negócios está mudando o jeito de como estes processos são modelados, executados e gerenciados. O SIG é importante em Inteligência de Negócios (IN) porque grande parte dos problemas em negócios inclui componentes espaciais significantes e o SIG permite que pessoas responsáveis por tomar decisões analisem seus dados espaciais de uma forma mais eficiente. Gerência de Relacionamento com o Consumidor, Planejamento de Recursos de Empresas, Gerenciamento de Suprimentos, e muitos outros são acrônimos para algumas soluções criadas para extrair e analisar informações de reservas de dados e permitir que pessoas responsáveis por tomar decisões desempenhem um alto nível de eficiência. Mas o dado em si não possui valor algum. Sem maneiras de visualização simples de integrar, demonstrar e analisar, é possível que resulte em uma grande quantidade de dados, mas nenhuma informação. De um ponto de vista particular, os dados geoespaciais e os mapas gerenciados por um software GIS corporativo representam um tipo de "linguagem" comum, que é reconhecida dentro e fora das fronteiras organizacionais. Esta "linguagem" possui, entre outras finalidades, o poder de integrar funções tradicionais e essenciais de negócios (Câmara *et al*, 1999).

CAPÍTULO 2 – DOT NET

A revolução da internet no final dos anos 90 representou uma dramática mudança na forma como indivíduos e organizações se comunicam entre si. Aplicativos tradicionais, como processadores de texto e pacotes de contas, são modelados como aplicativos únicos: eles oferecem ao usuário a capacidade de realizar tarefas utilizando um dado armazenado no sistema em que a aplicação reside e o processa. A maioria dos novos sistemas, em contraste, são criados baseados em um modelo de computação distribuída onde as aplicações trabalham em conjunto para fornecer serviços e expor funcionalidades. Como resultado, o papel principal da maioria dos novos sistemas está mudando para um modo de compartilhamento de informações (através de servidores e navegadores *web*), colaboração (através de e-mail e mensagem instantânea) e expressões individuais (através de *logs web*, também conhecido como Blogs e *e-zines* - revistas publicadas na web). Essencialmente, o papel básico do software está mudando, de fornecer funcionalidades discretas para fornecer serviços mais complexos. (Parihar *et al*, 2002)

O Framework .NET representa um conjunto de serviços e bibliotecas orientados a objeto, unificados, que adotam esta nova idéia de software voltado a rede. O Framework .NET, na verdade, é considerado a primeira plataforma criada dos pés a cabeça utilizando a Internet como idéia principal. (Parihar *et al*, 2002)

2.1 A Plataforma .NET

A plataforma .NET é hoje considerada a base onde a próxima geração de softwares será construída. A Microsoft investiu muito dinheiro em seu desenvolvimento, e está trabalhando muito para tornar esta nova tecnologia padrão. Uma grande lista de parceiros da Microsoft também anunciaram o suporte a ferramentas e componentes .NET, podendo ser visualizada no site da *Microsoft Development Network* (Microsoft Development Network, 2008)

A plataforma .NET é mais do que uma linguagem de programação, kit de desenvolvimento de software (SDK - *Software Development Kit*), ou até mesmo um sistema operacional. Ela oferece novos serviços poderosos, um novo formato binário, novas linguagens de manutenção, extensões de linguagens de manutenção para linguagens já existentes, entre muitos outros.

Utilizar todas estas novas ferramentas eficientemente não é possível sem um forte background da plataforma que irá potencializar sua aplicação.

O preceito por trás da plataforma .NET é que o mundo da computação está mudando, de um dos inúmeros computadores conectados a servidores através de uma rede, como a internet, até um onde todos os dispositivos inteligentes, computadores e serviços trabalham juntos para providenciar um grande número de vantagens ao usuário. A plataforma .NET é a resposta da Microsoft para os desafios que esta mudança trará para os desenvolvedores de software.

A plataforma .NET possui vários componentes. Servidores como BizTalk e SQL Server, também como serviços .NET My Services e seu primeiro componente visível, .Net Passport, estão sendo descritos por alguns como partes integrais da plataforma .NET. No entanto, para muitos de nós, o Framework .NET é o que vêm a nossa cabeça quando o .NET é mencionado. Nele inclui o Visual Studio.NET (VS.NET), A *Common Language Runtime* (CLR) .NET, e a *Base Class Libraries* (BCL) .NET. Os outros componentes podem ser indispensáveis por aplicações específicas, mas eles não são partes necessárias para todas as aplicações .NET.

Analisando a arquitetura como um todo, podemos dividir o .NET em três componentes primários (Turtschi *et al*, 2002). A Figura 2 demonstra a arquitetura da plataforma .NET.

- 1) **O Framework .NET:** uma plataforma de desenvolvimento de aplicações completamente nova.
 - A) **A Common Language Runtime (CLR):** Um ambiente de execução que cuida de alocação de memória, detecção de erros, e interação com os serviços do sistema operacional.
 - B) **A Base Class Library (BCL):** Uma coleção de componentes de programação e APIs (Application Program Interfaces)
 - C) **Dois alvos de desenvolvimento de alto-nível:** Um para aplicações Web (ASP.NET) e outra para aplicações Windows (Windows Forms).

- 2) **Diversos produtos .NET:** Várias aplicações da Microsoft baseadas no Framework .NET, incluindo novas versões do Exchange e SQL Server, que são consideradas como *Extensible Markup Language* (XML), integrada na plataforma .NET.

- 3) **Diversos serviços .NET:** Disponibilizados pela Microsoft para o uso no desenvolvimento de aplicações utilizando o Framework .NET. O projeto Hailstorm é considerado, na verdade, uma tentativa de juntar alguns dos mais cruciais serviços web através do nome e marca Microsoft.

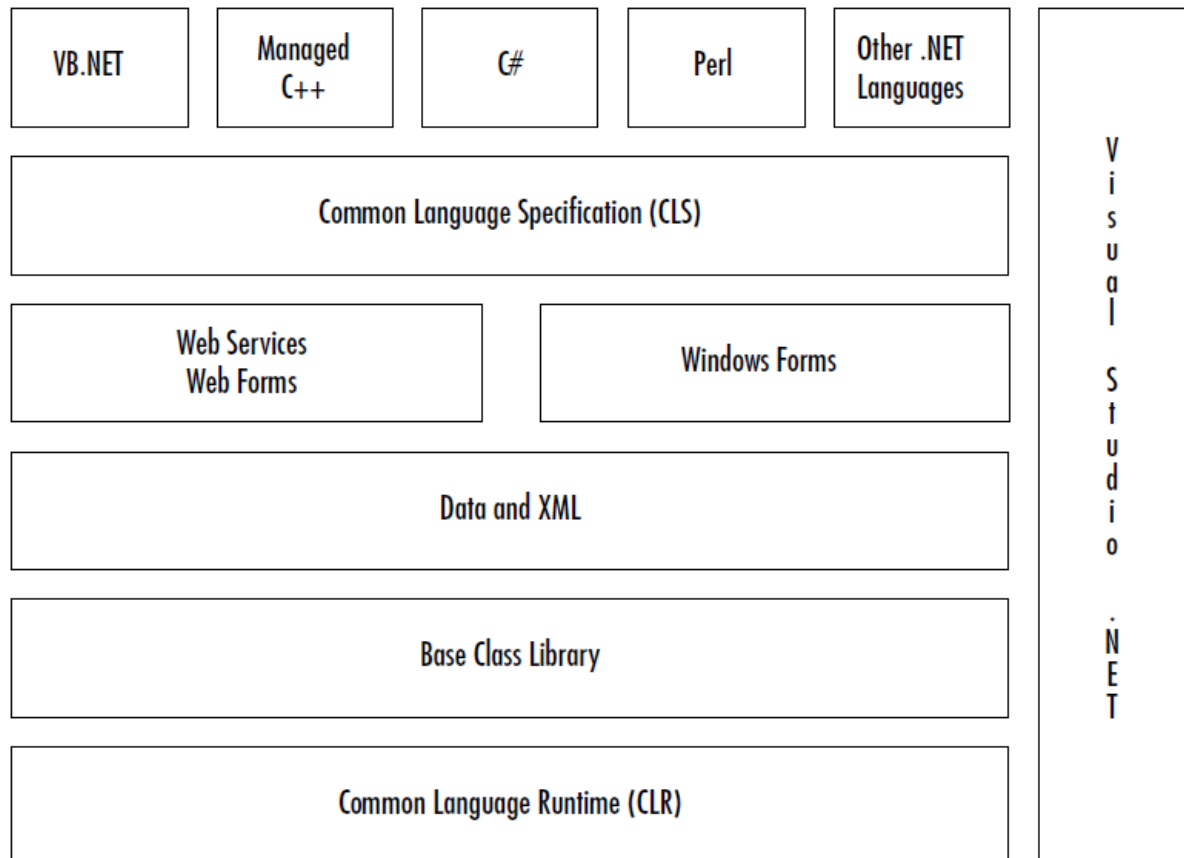


Figura 2. Arquitetura .NET (Turtschi *et al*, 2002)

As vantagens oferecidas pelo Framework .NET incluem ciclos mais curtos de desenvolvimento (re-uso de códigos, menos surpresas ao programar, suporte para várias linguagens de programação), mais facilidade de implantação, menos erros relacionados a tipos de dados, vazamento de memória reduzidos, graças ao coletor de lixo e, em geral, aplicações mais escaláveis e mais confiáveis. (Turtschi *et al*, 2002)

O Framework .NET trás uma grande quantidade de ferramentas que ajudam na elaboração de códigos para o próprio Framework .NET. A Microsoft oferece uma variedade de linguagens que já são compatíveis com o .NET, e o C# é uma delas. As novas versões do Visual Basic e Visual C++ também foram criadas para aproveitar as vantagens do Framework .NET, e uma versão do Jscript.NET já está a caminho.

O desenvolvimento de linguagens compatíveis com o .NET não está restrita a Microsoft. O grupo .NET da Microsoft publicou uma documentação mostrando como os fornecedores de linguagens podem fazer suas criações funcionarem com o .NET, e eles já estão fazendo linguagens como COBOL e Pearl compatíveis com o Framework .NET. Existem em torno de 20 linguagens ou mais sendo trabalhadas por outros fornecedores e instituições para em um futuro próximo também serem compatíveis com o Framework .NET. (Ferguson *et al*, 2002)

2.2 Desenvolvimento Web

O Framework .NET foi criado com um propósito: alimentar o desenvolvimento da Internet. Este novo atributo adicionado ao desenvolvimento da Internet é chamado de Serviços *Web* (*Web Services*). Podemos pensar em *Web Services* como sendo um *web site* que interage com programas, ao invés de pessoas. Ao invés de entregar páginas web, um *Web Service* pega uma solicitação formatada em XML, realiza uma determinada função, e finalmente retorna uma resposta a quem requisitou a solicitação através de uma mensagem XML.

O XML, ou *eXtensible Markup Language*, é uma linguagem auto-descritiva assim como HTML. XML, por outro lado, não possui tags pré-definidas, permitindo uma grande flexibilidade na representação de uma grande variedade de objetos. (Ferguson *et al*, 2002)

Uma aplicação típica para um *Web Service* seria o topo de um sistema de pagamento de uma empresa. Quando o usuário compra produtos através de uma loja na Internet ao navegar pela web, as informações da compra são enviadas para os *Web Services*, que totalizam todos os produtos, adicionam um registro as contas correspondentes no banco de dados, e então retornam uma resposta com um número de confirmação da compra. Este *Web Service* não apenas pode interagir com as páginas web, mas também pode interagir com outros *Web Services*, como os sistemas de pagamento de contas corporativas, por exemplo.

Para que o modelo *Web Service* sobreviva a evolução natural das linguagens de programação, ela deve incluir muito mais do que uma simples interface para a *Web*. O modelo *Web Service* também inclui protocolos, como por exemplo o protocolo SOAP (*Simple Object Access Protocol*), que habilitam aplicações para procurar *Web Services* disponíveis em uma LAN ou até mesmo na Internet. Este protocolo também habilita a aplicação a explorar os *Web Services* e determinar como se comunicar com eles, e também como trocar as informações.

Para habilitar a descoberta dos *Web Services*, foi estabelecida a *Universal Discovery, Description and Integration* (UDDI). Sendo assim, os *Web Services* agora podem ser registrados e procurados, baseados em informações chave como nome da empresa, tipo de serviço, e localização geográfica, conforme mostra o exemplo na Figura 3. (Ferguson *et al*, 2002)

2.3 O ambiente ASP.NET

A Internet foi originalmente desenvolvida para entregar conteúdo estático aos navegadores *Web*. Estas páginas *Web* nunca mudaram e foram as mesmas para todos os usuários que navegarem em sua localidade. *Active Server Pages* (ASP) foram criadas para tornar possível a criação de páginas dinâmicas baseadas em entradas feitas pelo usuário e interação com um *Web site*. Isto foi realizado com sucesso através da criação de scrips atrás das páginas *Web*, muita das vezes em VB Script.

O ASP.NET é uma melhora comparada ao ASP original, fornecendo agora o code-behind. Em ASP, o HTML e o script eram misturados em um único documento. Com o ASP.NET e o code-behind, o código e o HTML podem ser separados. Agora, quando a lógica de um *Web site* precisa ser alterada, não é necessário buscar entre centenas de linhas de código HTML para localizar o script que precisa ser modificado.

Assim como o *Windows Forms*, ASP.NET também suporta *Web Forms*. *Web Forms* permite que você "arraste e solte" controles em seus formulários, e fazer o *code-behind* da mesma forma que uma aplicação *Windows* típica.

Devido ao fato do ASP.NET utilizar o Framework .NET, ele também utiliza o compilador *just-in-time* (JIT). Páginas ASP tradicionais carregam com muita lentidão pois o código foi interpretado. O ASP.NET compila o código quando é instalado no servidor ou durante a primeira vez que é requisitado, o que resulta em um grande aumento de velocidade.

O processo chamado de compilação "just-in-time", ou simplesmente JIT, é a transformação de um código intermediário MSIL (Microsoft Intermediate Language) em código de máquina quando ele for executado pela primeira vez. A função do compilador JIT é traduzir o código MSIL genérico em código de máquina que poderá ser executado pela CPU. (Ferguson *et al*, 2002)

```

<businessEntity businessKey="ba744ed0-3aaf-11d5-80dc-002035229c64"
  operator="www.ibm.com/services/uddi"
  authorizedName="0100001QS1">
  <discoveryURLs>
    <discoveryURL
useType="businessEntity">http://www.ibm.com/services/uddi/uddiget?businessKey=BA744ED0-
3AAF-11D5-80DC-002035229C64</discoveryURL>
    </discoveryURLs>
  <name>XMethods</name>
  <description xml:lang="en">Web services resource site</description>
  <contacts>
    <contact useType="Founder">
      <personName>Tony Hong</personName>
      <phone useType="Founder" />
      <email useType="Founder">thong@xmethods.net</email>
    </contact>
  </contacts>
  <businessServices>
<businessService serviceKey="d5921160-3e16-11d5-98bf-002035229c64"
  businessKey="ba744ed0-3aaf-11d5-80dc-002035229c64">
  <name>XMethods Delayed Stock Quotes</name>
  <description xml:lang="en">20-minute delayed stock quotes</description>
  <bindingTemplates>
    <bindingTemplate bindingKey="d594a970-3e16-11d5-98bf-002035229c64"
  serviceKey="d5921160-3e16-11d5-98bf-002035229c64">
      <description xml:lang="en">SOAP binding for delayed stock quotes service</description>
      <accessPoint URLType="http">http://services.xmethods.net:80/soap</accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="uuid:0e727db0-3e14-11d5-98bf-002035229c64" />
      </tModelInstanceDetails>
    </bindingTemplate>
  </bindingTemplates>
  </businessService>
</businessServices>
</businessEntity>

```

Figura 3. Trecho de código de um registro Universal Discovery, Description and Integration.

2.4 C, C++ e C#

A linguagem de programação C# foi criada nos padrões das linguagens de programação C e C++. Nesta linguagem podemos notar seus poderosos recursos e a baixa curva de aprendizagem. Mas devido ao fato do C# ter sido construído dos pés a cabeça, alguns dos recursos mais onerosos, como os ponteiros, foram removidos.

A linguagem de programação C foi originalmente criada para o uso no sistema operacional UNIX. A mesma foi usada para criar inúmeras aplicações UNIX, incluindo um

compilador C, e foi eventualmente usado para escrever o próprio UNIX. Sua grande aceitação no mundo acadêmico expandiu e chegou ao mundo comercial, e fornecedores de software como a Microsoft e a Borland lançaram compiladores C para computadores pessoais. A API original do Windows foi criada para trabalhar com o código escrito em C, e até hoje os sistemas operacionais Windows continuam compatíveis com a linguagem.

Do ponto de vista de design, C não incluiu um detalhe que outras linguagens, como o Smalltalk, por exemplo, já tinham abraçado: o conceito de objeto. Codificação no estilo objeto poderia ser realizada usando C, mas a noção de um objeto não foi concretizada pela linguagem. Os objetos não eram uma parte necessária da mesma, portanto muitas pessoas não prestaram muita atenção a este paradigma de programação.

Depois que a noção do desenvolvimento da orientação objeto começou a ganhar reputação, ficou claro que o C precisava ser refinado para abraçar este novo jeito de programar. O C++ foi criado para incorporar esta nova abordagem. Ele foi criado para ser compatível com C, ou seja, todos os programas em C também seriam programas em C++ e poderiam ser compilados utilizando um compilador C++. O C++ adicionou suporte a classes (que são os "modelos" dos objetos), e permitiu uma grande geração de programadores C a pensarem em termos de objetos e seus comportamentos.

O C++ é um aperfeiçoamento do C, mas ainda possui algumas desvantagens. C e C++ podem dar um pouco de trabalho. Ao contrário de linguagens “*easy-to-use*”, como Visual Basic, por exemplo, C e C++ são muito "baixo nível" e necessitam que o programador codifique muito para que a aplicação rode sem graves problemas. É necessário escrever seu próprio código para tratar casos como gerenciamento de memória e detecção de erros. C e C++ podem resultar em aplicações muito poderosas, mas é preciso garantir que seu código funcione corretamente. Um pequeno problema pode fazer com que todo o programa colapse ou se comporte de forma inesperada. Devido a meta de criar o C++ compatível com o C, ele foi incapaz de sair da natureza baixo nível já caracterizada na linguagem C.

A Microsoft criou o C# mantendo grande parte da sintaxe do C e C++. Desenvolvedores que já estão familiarizados com estas linguagens podem pegar um código C# e sair codificando relativamente rápido. A grande vantagem do C#, no entanto, é que seus criadores escolheram não deixar a linguagem compatível com C e C++. Enquanto isso possa parecer uma má idéia, na verdade não é. O C# elimina as características que tornam o C e o C++ difíceis de se trabalhar. Pelo fato de todo código C ser também um código C++, o C++ teve que reter todas as dificuldades e deficiências encontradas no C. O C# está começando do zero e sem nenhum requisito de compatibilidade, podendo assim reter as vantagens de seus

antecessores e descartar as fraquezas que tornaram as vidas difíceis para os programadores de C e C++. (Ferguson *et al*, 2002).

Na Figura 4 podemos observar um exemplo de código implementado em diferentes linguagens, incluindo Visual Basic, C# e C++.

```

Visual Basic
Imports System

Public Class App
    Public Shared Sub Main()
        Dim o As Object = Nothing
        Dim p As Object = Nothing
        Dim q As New Object
        Console.WriteLine(Object.ReferenceEquals(o, p))
        p = q
        Console.WriteLine(Object.ReferenceEquals(p, q))
        Console.WriteLine(Object.ReferenceEquals(o, p))
    End Sub
End Class
' This code produces the following output:
'
' True
' True
' False
'
'

C#
using System;

class MyClass {

    static void Main() {
        object o = null;
        object p = null;
        object q = new Object();

        Console.WriteLine(Object.ReferenceEquals(o, p));
        p = q;
        Console.WriteLine(Object.ReferenceEquals(p, q));
        Console.WriteLine(Object.ReferenceEquals(o, p));
    }
}

/*

This code produces the following output.

True
True
False

*/

Visual C++
using namespace System;
int main()
{
    Object^ o = nullptr;
    Object^ p = nullptr;
    Object^ q = gcnew Object;
    Console::WriteLine( Object::ReferenceEquals( o, p ) );
    p = q;
    Console::WriteLine( Object::ReferenceEquals( p, q ) );
    Console::WriteLine( Object::ReferenceEquals( o, p ) );
}

/*

This code produces the following output.

True
True
False

*/

```

Figura 4. Exemplos de códigos em diferentes linguagens utilizando o método Object.ReferenceEquals (MSDN, 2008).

2.5 A IDE Visual Studio.NET

Além da poderosa plataforma .NET, a Microsoft apresentou uma nova versão do Visual Studio Suite, chamada Visual Studio.NET (VS.NET). Mesmo em suas versões Beta, o VS.NET oferece ao desenvolvedor ferramentas visuais poderosas para desenvolver diferentes tipos de aplicações na plataforma .NET.

Considerado como um ambiente de desenvolvimento completo, O VS.NET ajuda a acelerar o processo de criação e implantação de aplicações codificadas em qualquer das linguagens suportadas, incluindo o C#. Os componentes permanecem os mesmos independentes da linguagem, tornado muito fácil de modificar projetos e linguagens e manter os mesmos recursos no mesmo lugar. Além disso, com o IntelliSense, que auto-completa as tags, a codificação de rotinas são mais rápidas. (Ferguson *et al*, 2002)

CAPÍTULO 3 – DADOS REGIONAIS

3.1 Modelagem de Dados

Informações regionais precisas são essenciais para a criação de um Sistema de Informação Geográfica. O grande problema na obtenção deste dado é que, mesmo existindo uma boa quantidade de dados disponíveis, não existe um formato padrão, tornando a busca pouco intuitiva e dificultando a tomada de decisões.

Outro problema que dificulta o gerenciamento destas informações é o grande volume de dados a serem processados e a heterogeneidade dos tipos de dados e ambientes (software e hardware).

Quando se faz necessária uma busca específica de informações agropecuárias, normalmente envolve parâmetros muito específicos que dificilmente estarão disponíveis sem a necessidade de integrar diferentes tipos de dados. Cada tipo de informação normalmente está presente em diferentes documentos, planilhas, banco de dados, arquivos, entre outros, dificultando ainda mais sua integração e recuperação. (Simon e Tomasic, 1997)

Ao utilizarmos um SIG para extração e organização destes dados, podemos disponibilizar ao usuário toda a liberdade de parâmetros possíveis para busca dos dados. Uma importante abordagem utilizada para organizar estes dados dentro de um SIG é o uso de metadados.

Metadados são chamados de "dados sobre dados". Eles descrevem o conteúdo, qualidade, condição, e outras características do dado com o objetivo de ajudar o usuário a localizá-los e entendê-los. A criação dos metadados possui três objetivos principais (FGDC, 2001). O primeiro objetivo é organizar e manter um investimento feito por uma empresa nos dados. Quando um funcionário muda ou o tempo passa, as informações sobre os dados de uma empresa podem se perder. Mais tarde, os funcionários podem possuir pouco entendimento sobre o conteúdo e uso de um banco de dados digital e podem chegar à conclusão que não poderão confiar nos resultados gerados por estes dados. Descrições completas do conteúdo e a precisão dos dados geográficos resultarão em um uso mais apropriado dos dados.

O segundo objetivo é fornecer informações para catálogos de dados e *clearinghouses*. Em grande parte dos casos, o SIG necessita de uma integração de dados de diferentes tipos de fontes, e poucas empresas podem pagar a criação de todos os dados que

elas necessitam. Com isto em vista, dados criados por uma empresa ou organização podem ser úteis a outras. Fazendo com que os metadados sejam disponíveis através de catálogos de dados e *clearinghouses*, empresas podem buscar dados para usar, parceiros para compartilhar coleções de dados e serviços de manutenção, e consumidores para seus dados.

Finalmente, o terceiro objetivo dos metadados é fornecer informações para ajudar a transferência de dados. Metadados devem acompanhar a transferência de um registro. Ele ajuda na organização, recebendo o processo dos dados e os interpretando, incorporando-os em suas categorias e atualizando catálogos com as novas informações.

3.2 Disponibilidade de dados agropecuários em Marília e região

Atualmente, a busca de dados específicos relacionados a agropecuária de Marília e região é pouco intuitiva por alguns motivos:

- 1) As informações disponibilizadas pelo governo do estado de São Paulo estão separadas por cidades e atributos em diferentes arquivos, conforme mostrado na Figura 5. Não existe, deste modo, a possibilidade de associar os dois parâmetros numa busca. Por exemplo, se para o usuário não interessa a cidade, e apenas onde possui poucas plantações de laranja, ele terá que manualmente selecionar as cidades através de uma lista não ordenada.
- 2) Pelo fato da seleção dos dados precisar ser feita através de humanos, conforme mencionado no item acima, há uma preocupação com a confiabilidade dos dados. Não é possível afirmar que os dados selecionados por uma pessoa são realmente os melhores selecionados para aquela busca. Basta apenas uma linha mal interpretada do documento para atrapalhar na tomada de decisões de uma empresa.
- 3) Uma informação não precisa apenas ser confiável para ser considerada relevante. Dentre outras características, ela também precisa estar disponível no momento certo. Quando existe a necessidade de verificar diferentes documentos e compará-los, principalmente se tratando de arquivos com grande quantidade de dados, o tempo gasto nesta tarefa pode trazer atrasos indesejáveis.

Ao analisar estes itens, podemos perceber o quanto a forma que os dados são disponibilizados ao usuário é importante. Pelo fato de uma empresa não precisar passar por todo este processo de pesquisa pode significar um incentivo e, conseqüentemente, despertar

um interesse dela pela região, trazendo vantagens como avanço tecnológico, criação de novos empregos, entre outras.

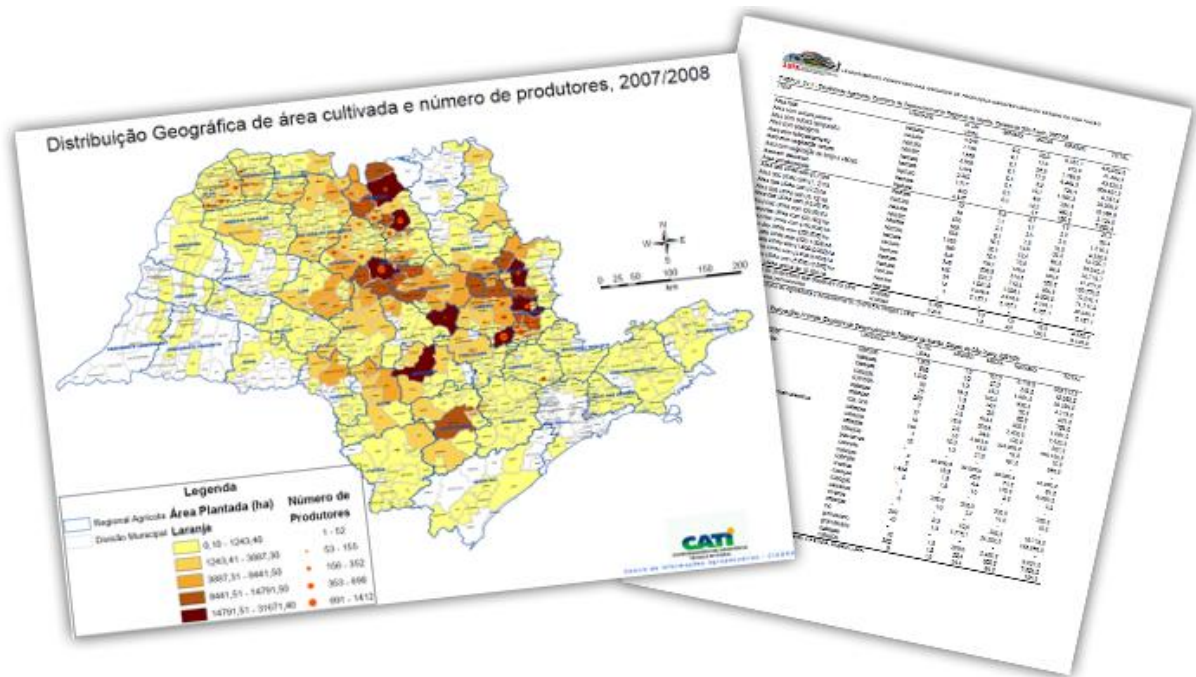


Figura 5. Exemplos de documentos com dados agropecuários da região de Marília (LUPA, 2008).

Desta maneira, o objetivo do presente trabalho é implementar um ambiente que possibilite a extração de atributos referentes à agronomia e pecuária de regiões pré-estabelecidas. Com base nos mapas provenientes de sensoriamento remoto, o usuário deve ser capaz de selecionar determinados pontos georreferenciados das imagens de satélite e obter dados combinados ou isolados da zona, sobre um determinado aspecto previamente conhecido, tais como informações de vegetação, relevo, cultura e criações. Finalmente, pretende-se tornar intuitiva a busca por informações agropecuárias, baseando-se no processo visual automatizado descrito.

CAPÍTULO 4 – METODOLOGIA

O formato de visualização dos dados existente até então não é de fácil interpretação e dificulta significativamente sua compreensão. Com um ambiente mais interativo e dinâmico, o usuário terá acesso a informação desejada de uma maneira mais exata, destacando apenas a região ou ponto específico que lhe interessa.

Este ponto específico será escolhido através de uma imagem da região de Marília-SP, proveniente de sensoriamento remoto (Embrapa, 2009), conforme ilustrado na Figura 6, fazendo assim com que o usuário realmente veja o ponto que lhe interessa e analise os dados com a imagem.



Figura 6. Mapa de Marília e Região utilizado no SIG.

4.1 – Introdução

Para o desenvolvimento do projeto, serão utilizadas técnicas de Computação Gráfica implementadas sob a linguagem *ASP.NET* com o uso da IDE *Microsoft Visual Studio*. Desta forma, os modelos de dados, representados sob a forma de mapas provenientes de sensoriamento remoto, poderão ser explorados mediante diversos aspectos geográficos e agropecuários, visando a extração de atributos e a busca por informações relevantes,

combinadas ou não. O diagrama representado pela Figura 7 ilustra a visão geral do SIG desenvolvido.

A primeiro momento foi desenvolvido um banco de dados em MySQL para atender as necessidades de armazenamento de dados geográficos. O software é capaz de armazenar as informações e recuperá-las através de buscas ao banco.

Na primeira etapa, o sistema carrega o mapa em relevo da região a ser estudada, a fim de que o usuário possa interagir com regiões disponíveis através das ferramentas disponíveis no SIG. À *priori*, as regiões serão pré-determinadas, a fim de trabalhar em conjunto com o banco de dados.

Na segunda etapa, o usuário deve selecionar a área de interesse para coletar as informações necessárias naquele momento. Seja pela interação sobre a interface gráfica (mapa em relevo) ou por um conjunto de atributos, o sistema irá selecionar a coordenada do ponto escolhido e buscar suas informações no banco de dados (terceira etapa). Na quarta e última etapa, o sistema irá exibir ao usuário as informações coletadas do banco de dados sobre a coordenada ou atributo em questão.

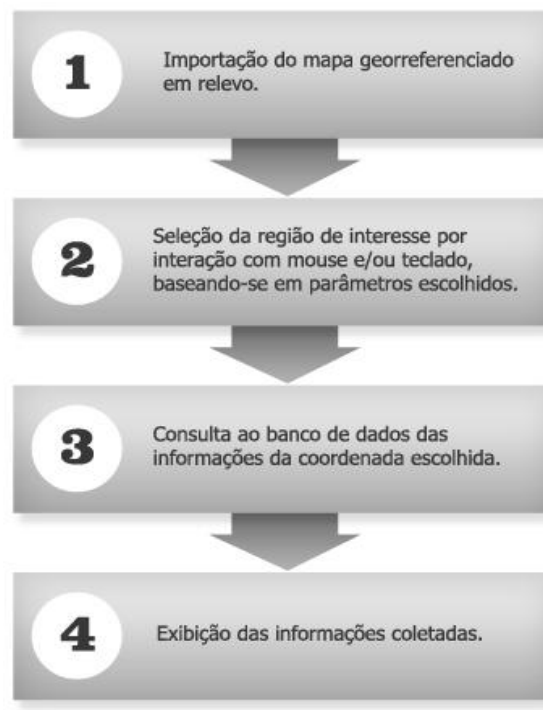


Figura 7. Visão geral do funcionamento do SIG proposto.

4.2 Funcionalidades

A grande vantagem de utilizar um Sistema de Informação Geográfica para realizar a extração de dados geográficos é a existência de vários parâmetros para se realizar uma busca dos dados. O sistema proposto, por exemplo, conforme mostra o digrama de funcionalidades representado na Figura 8, visa buscar dados de agricultura e pecuária de Marília e região, e o usuário poderá buscar os dados desejados das seguintes maneiras:

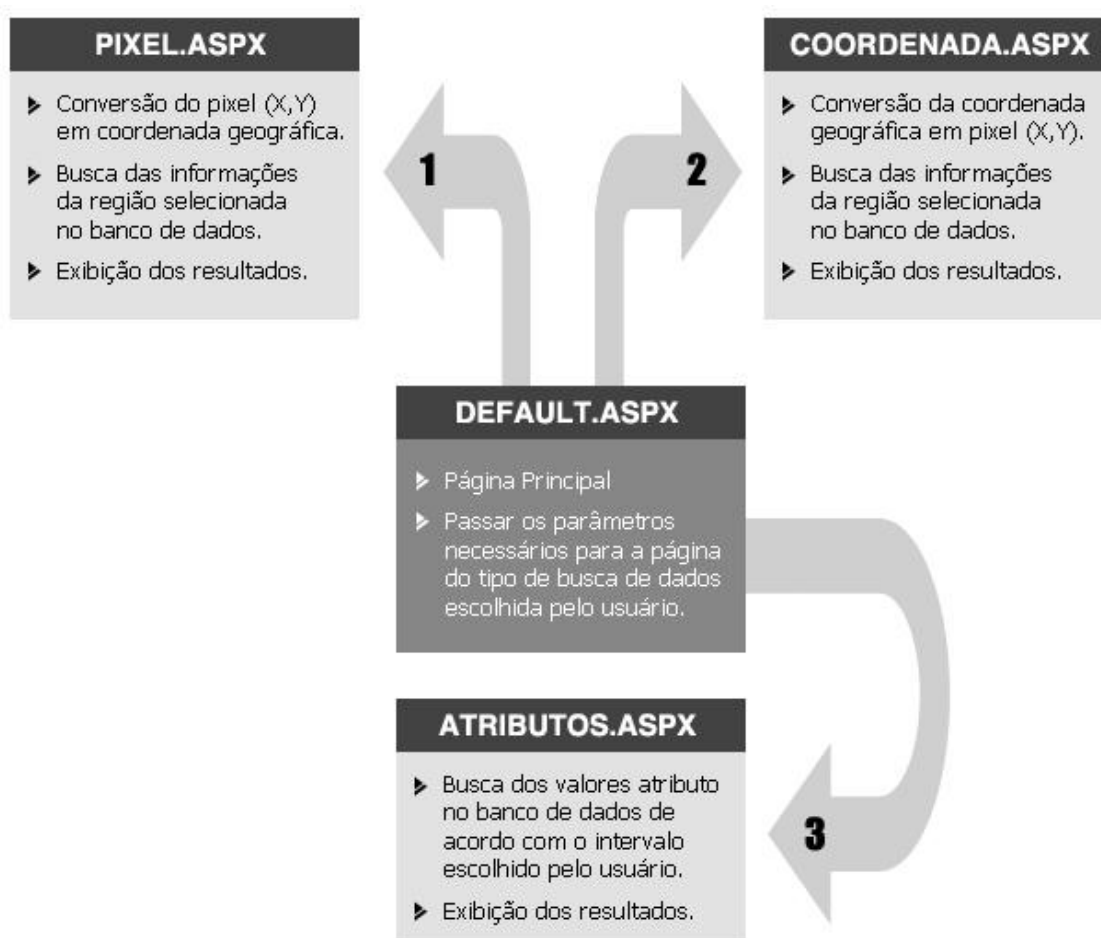


Figura 8. Diagrama de Funcionalidades do Sistema.

4.2.1 – Extração de dados a partir de mapa geo-referenciado

A primeira opção disponível no sistema é a busca dos dados a partir de um ponto específico em um mapa geo-referenciado. Neste modo, o usuário deve simplesmente clicar em qualquer ponto do mapa para coletar as informações agropecuárias da região que o representa.

O mapa de Marília e região é dividido em 28 áreas iguais (4x7), onde duas delas são sub-dividas em mais duas áreas cada, resultando em 30 regiões diferentes. Cada região representa, no máximo, uma cidade (Figura 9).

coord_id	coord_x1	coord_y1	coord_x2	coord_y2	coord_regnome
1	0	0	100	102	Quintana
2	99	0	200	102	Pompéia
3	199	0	300	102	NÃO HÁ CIDADES NESTA ÁREA
4	299	0	400	102	NÃO HÁ CIDADES NESTA ÁREA
5	399	0	500	102	Júlio Mesquita
6	499	0	600	102	Álvaro de Carvalho
7	599	0	699	102	NÃO HÁ CIDADES NESTA ÁREA
8	0	102	100	204	NÃO HÁ CIDADES NESTA ÁREA
9	99	102	200	204	NÃO HÁ CIDADES NESTA ÁREA
10	199	102	300	204	Oriente
11	299	102	400	204	Marília
12	399	102	500	204	Vera Cruz
13	499	102	600	204	Garça
14	599	102	699	204	NÃO HÁ CIDADES NESTA ÁREA
15	0	204	100	306	Oscar Bressane
16	99	204	200	306	NÃO HÁ CIDADES NESTA ÁREA
17	199	204	300	306	NÃO HÁ CIDADES NESTA ÁREA
18	299	204	400	306	NÃO HÁ CIDADES NESTA ÁREA
19	399	204	500	306	NÃO HÁ CIDADES NESTA ÁREA
20	499	204	600	306	NÃO HÁ CIDADES NESTA ÁREA
21	599	204	699	255	Gália
22	0	306	100	404	NÃO HÁ CIDADES NESTA ÁREA
23	99	306	200	404	Echaporã
24	199	306	300	404	NÃO HÁ CIDADES NESTA ÁREA
25	299	306	400	404	Ocaçu
26	399	306	464	404	Lupércio
27	499	306	600	404	NÃO HÁ CIDADES NESTA ÁREA
28	599	306	699	404	Lucianópolis
29	599	255	699	306	Fernão
30	464	306	500	404	Alvinlândia

Figura 9. Tabela de “metadados” do SIG incluindo a divisão das áreas no mapa.

O sistema realiza esta operação através do valor das coordenadas X e Y da imagem (pixels) e associando a mesma com a coordenada geográfica real. Levando em conta que o par

(0,0) da imagem representa a coordenada $50^{\circ} 22' 30''$ W (longitude) e 22° S (latitude), e o par (700,405), que representa o último pixel da imagem, representa a coordenada $49^{\circ} 30'$ W (longitude) e $22^{\circ} 30'$ S (latitude), o sistema calcula o valor das coordenadas a partir de cálculos envolvendo estas informações iniciais (Figura 10).

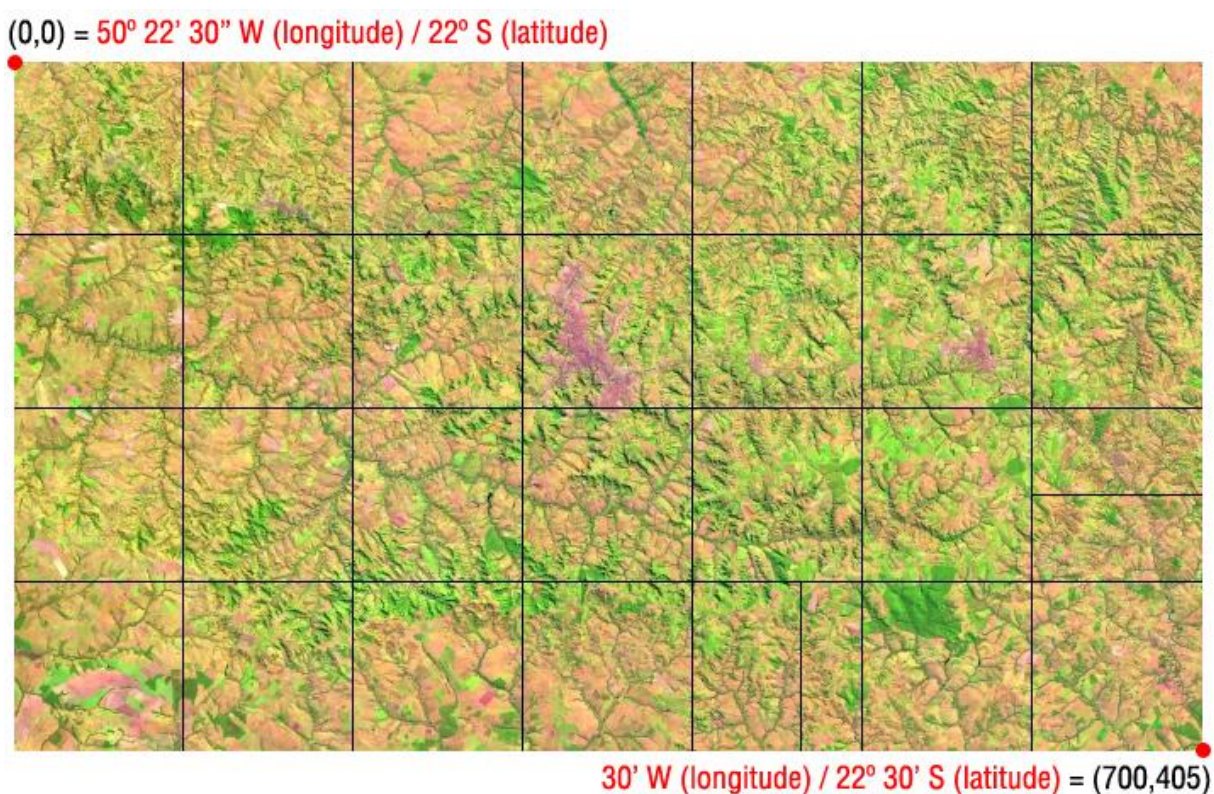


Figura 10. Mapa dividido em regiões de acordo com as cidades (EMBRAPA, 2009)

Todas as coordenadas são transformadas em segundos para a comparação com os pixels. Por exemplo, o pixel (0,0) representa, em coordenada, (181350, 79200).

Quando o usuário clica no mapa, é passado os valores (X,Y) do ponto escolhido como variável de sessão para o arquivo Pixel.aspx, onde é feita a busca da região no banco de dados, e o cálculo da coordenada geográfica. Quando o ponto escolhido pertencer a uma região que represente uma cidade, é retornado os dados de agricultura e pecuária da mesma após uma busca a tabela de dados do banco de dados.

4.2.2 Extração de dados a partir de coordenada geográfica

Quando é necessário efetuar uma busca a partir de uma coordenada geográfica da região, o usuário poderá usar esta função do sistema. Basta inserir nos campos os valores da

latitude e longitude e clicar em “Buscar”. Caso a coordenada geográfica não esteja dentro do limite da região abordada pelo sistema, será mostrada uma mensagem de erro.

Quando o botão de busca é pressionado, é passado como variável de sessão os seis valores informados pelo usuário. Fazendo o processo contrário da primeira opção, neste modo a coordenada é primeiramente transformada em segundos para depois ser convertida em pixels. Deste modo, é possível buscar as informações da região através dos mesmos parâmetros da função anterior e utilizar a mesma tabela do banco de dados, conforme mostrado na Figura 11.

4.2.3 – Extração de dados a partir de atributos geográficos e agropecuários

Nesta terceira e última opção de busca de dados, o usuário pode coletar os dados a partir de um determinado atributo listado em uma *DropDownList* com todas as opções possíveis. Além deste parâmetro, o usuário também deve informar um intervalo (valor mínimo e valor máximo) numérico para a busca dos dados.

Neste tipo de busca também podem ser utilizados como parâmetro medidas e aspectos geográficos, como por exemplo: área total, área complementar, área com reflorestamento, área com certo tipo de vegetação, entre outros.

Funcionalmente, isto é possível passando como parâmetro o nome do atributo e os valores do intervalo desejado. A partir destas informações, é realizada uma busca na tabela de dados geográficos do banco de dados e os resultados correspondentes são exibidos ao usuário, conforme mostrado na Figura 12.

Conforme podemos observar no código, os valores informados pelo usuário são comparados com os valores dos atributos presentes no banco de dados. Sendo assim, o valor real do atributo que está sendo usado na busca deve ser maior que o valor mínimo informado pelo usuário e, conseqüentemente, menor que o valor máximo informado pelo mesmo.

O sistema também permite que o usuário deixe as informações de valor mínimo e máximo em branco. Deixando o valor mínimo em branco, ele será automaticamente atribuído a zero e, o valor máximo quando não preenchido, será o valor máximo de cada atributo. Caso os dois estejam desta forma, serão listadas, conseqüentemente, todas as cidades onde existe qualquer incidência do atributo escolhido pelo usuário e o seu valor total.

Pixel.aspx.cs

```

protected void Page_Load(object sender, EventArgs e)
{
    int pixX = Convert.ToInt32(Session["X"]);
    int pixY = Convert.ToInt32(Session["Y"]);

    CoordX.Text = Convert.ToString(pixX);
    CoordY.Text = Convert.ToString(pixY);

    double num, temp, grau, min, seg;
    num = 181350 - (Convert.ToDouble(Session["X"])*4.5);

    temp = Math.Floor(num / 60);
    seg = Math.Floor(num % 60);
    min = (temp % 60);
    grau = Math.Floor(temp / 60);

    XSeg.Text = Convert.ToString(seg);
    XMin.Text = Convert.ToString(min);
    XGrau.Text = Convert.ToString(grau);

    num = 79200 + (Convert.ToDouble(Session["Y"])*4.6);

    temp = Math.Floor(num / 60);
    seg = Math.Floor(num % 60);
    min = (temp % 60);
    grau = Math.Floor(temp / 60);

    YSeg.Text = Convert.ToString(seg);
    YMin.Text = Convert.ToString(min);
    YGrau.Text = Convert.ToString(grau);

    String ConStr = ConfigurationManager.
    ConnectionStrings["SIGConnectionString"].ConnectionString;
    SqlConnection con = new SqlConnection(ConStr);

    string SQL = "SELECT coord_id FROM COORD WHERE ((coord_x1 < @coord_x1)
    AND (coord_x2 > @coord_x2) AND (coord_y1 < @coord_y1) AND (coord_y2 > @coord_y2))";
    SqlCommand cmd = new SqlCommand(SQL, con);
    cmd.Parameters.Add(new SqlParameter("@coord_x1", pixX));
    cmd.Parameters.Add(new SqlParameter("@coord_x2", pixX));
    cmd.Parameters.Add(new SqlParameter("@coord_y1", pixY));
    cmd.Parameters.Add(new SqlParameter("@coord_y2", pixY));
    con.Open();
    SqlDataReader dr = cmd.ExecuteReader(CommandBehavior.Default);
    dr.Read();
    IDCidade.Text = dr["coord_id"].ToString();
}

```

Figura 11. Trecho de código demonstrando a conversão do valor em pixel em coordenada geográfica e a extração de dados a partir do mapa geo-referenciado.

Coordenada.aspx.cs

```

protected void Page_Load(object sender, EventArgs e)
{
    double num, grau, min, seg;
    int varx, vary;

    XGrau.Text = Convert.ToString(Session["LonG"]);
    XMin.Text = Convert.ToString(Session["LonM"]);
    XSeg.Text = Convert.ToString(Session["LonS"]);
    YGrau.Text = Convert.ToString(Session["LatG"]);
    YMin.Text = Convert.ToString(Session["LatM"]);
    YSeg.Text = Convert.ToString(Session["LatS"]);
    seg = Convert.ToDouble(Session["LonS"]);
    min = Convert.ToDouble(Session["LonM"]);
    grau = Convert.ToDouble(Session["LonG"]);

    min = grau * 60 + min;
    seg = min * 60 + seg;
    num = Math.Floor((181350 - seg) / 4.5);
    CoordX.Text = Convert.ToString(num);

    seg = Convert.ToDouble(Session["LatS"]);
    min = Convert.ToDouble(Session["LatM"]);
    grau = Convert.ToDouble(Session["LatG"]);
    min = grau * 60 + min;
    seg = min * 60 + seg;
    num = Math.Abs(Math.Floor((79200 - seg) / 4.6));
    CoordY.Text = Convert.ToString(num);
    varx = Convert.ToInt32(CoordX.Text);
    vary = Convert.ToInt32(CoordY.Text);

    String ConStr = ConfigurationManager.
ConnectionStrings["SIGConnectionString"].ConnectionString;
    SqlConnection con = new SqlConnection(ConStr);

    string SQL = "SELECT coord_id FROM COORD WHERE ((coord_x1 < @coord_x1)
AND (coord_x2 > @coord_x2) AND (coord_y1 < @coord_y1) AND (coord_y2 > @coord_y2))";
    SqlCommand cmd = new SqlCommand(SQL, con);
    cmd.Parameters.Add(new SqlParameter("@coord_x1", varx));
    cmd.Parameters.Add(new SqlParameter("@coord_x2", varx));
    cmd.Parameters.Add(new SqlParameter("@coord_y1", vary));
    cmd.Parameters.Add(new SqlParameter("@coord_y2", vary));
    con.Open();
    SqlDataReader dr = cmd.ExecuteReader(CommandBehavior.Default);
    dr.Read();
    IDCidade.Text = dr["coord_id"].ToString();
}

```

Figura 12. Trecho de código demonstrando a conversão da coordenada geográfica em pixel e a extração de dados a partir da coordenada geográfica.

CAPÍTULO 5 – RESULTADOS

Baseado nas funcionalidades descritas na metodologia, podemos observar os seguintes resultados para cada método de busca de dados:

5.1 Extração de dados a partir de atributos agropecuários

A partir do mapa oferecido ao usuário na página principal do sistema, o mesmo pode requisitar os dados de um determinado ponto, podendo ou não pertencer a uma cidade. Ao clicar neste ponto, o programa abre uma nova página com o ID da área escolhida, o nome da cidade que ela representa (se houver alguma cidade nesta área), a coordenada geográfica real do ponto escolhido e, finalmente, os dados geográficos e agropecuários da região, conforme ilustrado na Figura 13.

5.2 Extração de dados a partir de coordenada geográfica

Apesar de serem duas funcionalidades completamente diferentes no ponto de vista computacional, a maneira como o usuário vê os resultados é a mesma. O que difere, ocasionalmente, é o atributo passado para a extração das mesmas.

Os parâmetros informados para o usuário será a coordenada geográfica pura, tornando desnecessário, neste caso, o uso do mapa. Mesmo assim, o usuário é informado novamente da coordenada geográfica escolhida na tela de resultados em conjunto com o ID da área escolhida, o nome da cidade que ela representa (se houver alguma cidade nesta área), e os dados geográficos e agropecuários da região que representa a coordenada escolhida.

Como exemplo, o usuário requisitou informações sobre a coordenada geográfica 49° 42' W (longitude) e 22° 4' 40" S (latitude), o que representa a cidade de Álvaro de Carvalho. Os resultados podem ser vistos na Figura 14.

5.3 Extração de dados a partir de atributos geográficos e agropecuários

Neste caso, é levada em conta a necessidade de o usuário pesquisar a localização geográfica de onde há uma ocorrência de um determinado atributo, o contrário que foi realizado nas duas primeiras funcionalidades.

ID	CIDADE			
5	Júlio Mesquita	49	° 50	' 24 " W - Longitude
		22	° 3	' 17 " S - Latitude

TIPO DE DADO	UNIDADE	VALOR TOTAL
Área total	hectare	11688,40
Área com cultura perene	hectare	1534,80
Área com cultura temporária	hectare	359,30
Área com pastagens	hectare	8070,30
Área com vegetação mata	hectare	843,20
Área com vegetação de várzea	hectare	310,30
Área com reflorestamento	hectare	81,30
Área em descanso	hectare	362,70
Área complementar	hectare	126,50
Altura média	metros	520,00
Bovinocultura de corte	cabeças	9281,00
Bovinocultura mista	cabeças	1016,00
Asininos e muares	cabeças	6,00
Equinocultura	cabeças	804,00
Ovinocultura	cabeças	88,00
Suinocultura	cabeças	48,00
Braquiária	hectare	8060,30
Laranja	hectare	939,20
Café	hectare	457,40
Cana-de-açúcar	hectare	304,00
Eucalipto	hectare	81,30
Milho	hectare	55,30
Tangor	hectare	53,70
Tangerina	hectare	35,00
Banana	hectare	16,00
Abacate	hectare	11,00
Goiaba	hectare	11,00
Colonião	hectare	10,00
Manga	hectare	4,50
Urucum (ou urucu)	hectare	4,00
Limão	hectare	3,00

Figura 13. Resultado da funcionalidade de busca através do mapa geo-referenciado.

Primeiramente são exibidas as informações sobre a vegetação do local, seguidas de pecuária e, por fim, agricultura. Na coluna do meio é mostrada a unidade de medida do valor total, disponível na terceira coluna.

ID	CIDADE			
6	Álvaro de Carvalho	49	° 42	' 0 " W - Longitude
		22	° 4	' 40 " S - Latitude

TIPO DE DADO	UNIDADE	VALOR TOTAL
Área total	hectare	16502,70
Área com cultura perene	hectare	1460,50
Área com cultura temporária	hectare	413,20
Área com pastagens	hectare	12272,50
Área com vegetação mata	hectare	540,69
Área com vegetação capoeira	hectare	121,82
Área com vegetação cerrado	hectare	0,41
Área com vegetação de várzea	hectare	13,00
Área com reflorestamento	hectare	110,67
Área em descanso	hectare	2,90
Área complementar	hectare	70,90
Altura média	metros	627,00
Bovinocultura de corte	cabeças	15086,00
Bovinocultura de leite	cabeças	387,00
Bovinocultura mista	cabeças	248,00
Suinocultura	cabeças	147,00
Braquiária	hectare	12271,30
Café	hectare	1360,00
Amendoim	hectare	363,00
Eucalipto	hectare	302,70
Milho	hectare	55,00
Tangerina	hectare	45,90
Limão	hectare	41,60
Melancia	hectare	16,90
Cana-de-açúcar	hectare	16,60
Seringueira	hectare	11,00
Tomate rasteiro	hectare	4,80
Coco-da-baia	hectare	2,00
Capim-napier (ou capim-elefante)	hectare	1,20

Figura 14. Resultado da funcionalidade de busca através de coordenada geográfica.

Da mesma forma que a Figura 13, as informações sobre a vegetação do local são exibidas na primeira coluna, seguidas de pecuária e, por fim, agricultura. Na coluna do meio é mostrada a unidade de medida do valor total, disponível na terceira coluna.

Será informado ao sistema o tipo de atributo escolhido para a busca e um intervalo de valores para o mesmo.

Quando o usuário clica no botão de busca, o banco de dados pega o item selecionado e os valores numéricos, e busca no banco de dados. Caso o item for o mesmo do que está sendo comparado no banco de dados e seu respectivo valor estiver entre o valor mínimo e o valor máximo fornecido pelo usuário, serão listadas (em ordem decrescente por valor total) a cidade onde existe esta ocorrência, o atributo, sua unidade de medida e seu valor total.

Por exemplo, uma empresa possui o interesse em saber quais são os maiores produtores de cana-de-açúcar da região. Sendo assim, o usuário faz uma busca para determinar quais cidades possuem entre mil e dez mil hectares de plantação de cana-de-açúcar (Figura 15).

CIDADE	ATRIBUTO	UNIDADE	TOTAL
Ocaçu	Cana-de-açúcar	hectare	1627,70
Lucianópolis	Cana-de-açúcar	hectare	1734,00
Marília	Cana-de-açúcar	hectare	2463,80
Oriente	Cana-de-açúcar	hectare	3063,20
Quintana	Cana-de-açúcar	hectare	5155,60
Pompéia	Cana-de-açúcar	hectare	8161,90
Echaporã	Cana-de-açúcar	hectare	8329,30

Figura 15. Resultado da busca por atributo “cana-de-açúcar”, passando como limite de valores mil e dez mil hectares.

A partir do valor mínimo e máximo informado pelo usuário, as ocorrências do atributo escolhido foram listadas na tela. Estas informações são exibidas em ordem crescente a partir do valor total do atributo, localizado na quarta coluna. A primeira informa o nome da cidade, a segunda o nome do atributo, e a terceira coluna contém a unidade de medida do dado.

CAPÍTULO 6 – CONCLUSÕES

A busca de informações em geral de forma mais eficiente possui um grande valor para empresas de todos os ramos. Devido a esta necessidade, podemos notar o crescente número de Sistemas de Informação Geográfica disponíveis no mercado com a finalidade de auxiliar estas empresas a extrair, organizar e interpretar seus dados.

No ramo de agropecuária, esta necessidade não poderia ser diferente. Além da grande quantidade de dados e tipos de dados, estas informações devem estar muito bem organizadas. Estes aspectos fornecem ao usuário uma forma muito eficiente de buscar informações envolvendo diversos tipos de dados que, manualmente, envolveriam alguns riscos e muito mais tempo para associar.

Estas informações, hoje em dia, estão disponíveis para o usuário através de documentos, separados por cidades ou atributos, tornando impossível a associação dos dados sem a participação de um ser humano para realizar esta tarefa. Além de possuir uma baixa objetividade, este processo está propício a erros, colocando em risco a tomada de decisões.

Um dos principais objetivos na criação deste SIG é auxiliar na tomada de decisões. No sistema desenvolvido abrangendo a região de Marília, o usuário tem a liberdade de pesquisar atributos geográficos a partir de um ponto do mapa, sua coordenada geográfica, ou até mesmo pelos seus atributos. Deste modo, uma empresa interessada em se instalar na região terá esta parte do processo já resolvida. Com este incentivo, o resultado é o auxílio no desenvolvimento regional com todas as vantagens que a instalação de novas empresas trará a região.

Futuramente, este projeto poderá evoluir para abranger novas regiões. Para isso, seria interessante a implementação de um banco de dados mais específico para atributos geográficos, por exemplo.

Além disso, um SIG não precisa trabalhar de forma individual. É possível a associação com outros sistemas, possibilitando assim uma maior quantidade e qualidade de dados. Deste modo, o usuário terá a disposição uma maior quantidade de funções e novos parâmetros para a busca de informações.

Assim, conclui-se que a criação de um SIG para cada região do país e, finalmente, a integração de todos estes sistemas, poderia auxiliar e muito no desenvolvimento de pequenos e grandes negócios. Além disso, pode contribuir em aspectos como controle ambiental, organização de sensores e distribuição de terras.

REFERÊNCIAS BIBLIOGRÁFICAS

- Câmara G. *et al* (1999). **Geoprocessamento: Teoria e Aplicações**, 3-29 - 3,33. Disponível em <<http://www.dpi.inpe.br/gilberto/livro/>>
- Carr, M. H., e Zwick, P. D. (2007). **Smart land use analysis: The LUCIS Model**. Redlands, CA: ESRI Press.
- Castle, G. H. (1993). **Profiting from a geographic information system**. Ft. Collins, CO: GIS World.
- Chapin, F. S. (1965). **Urban land use planning** (2^a ed.). Champaign, IL: University of Illinois Press.
- Chapin, F. S. e Kaiser, E. J. (1979). **Urban land use planning** (3^a ed.). Champaign, IL: University of Illinois Press.
- Chrisman, N. R. (1998). Academic origins of GIS. Editora Foresman, **The history of geographic information systems: Perspectives from the pioneers** (pp. 33–43). Upper Saddle River, NJ: Prentice Hall.
- Chrisman, N. R. (2006). **Charting the unknown: How computer mapping at Harvard became GIS**. Redlands, CA: ESRI Press.
- Dueker, K. J. (1987). **Geographic information systems and computeraided mapping**. *Journal of the American Planning Association*, 53 (3), 383–390.
- Drummond, W. J. e French, S. P. (2008). **The Future of GIS in Planning, Converging Technologies and Diverging Interests**. *Journal of the American Planning Association*, Vol. 74, No. 2.
- Embrapa Monitoramento por Satélite, Brasil em Relevo (2009). Disponível em: <<http://www.relevobr.cnpem.embrapa.br/>>
- Enalta Inovações Tecnológicas para Agricultura (2008). **PIMS-SIG Agrícola**. Disponível em: <http://www.enalta.com.br/produtos_detalhes.php?categoria=9&atividade=12&css=0&produto=110>
- Federal Geographic Data Committee (FGDC) (2001). **Content Standard for Digital Geospatial Metadata Workbook**. Reston, VA, Federal Geographic Data Committee.

- Ferguson J., Patterson B. e Boutquin, P. (2002). **C# Bible**, Wiley Publishing, 1-17
- Ingmire, T. J., e Patri, T. (1971). **An early warning system for regional planning**. Journal of the American Institute of Planners, 37 (6), 403–410.
- Levantamento Censitário das Unidades de Produção Agropecuária do Estado de São Paulo (LUPA) (2008). Disponível em: <<http://www.cati.sp.gov.br/projetolupa/index.php>>
- Lyle, J., e von Wodtke, M. (1974). **An information system for environmental planning**. Journal of the American Institute of Planners, 40 (6), 394–413.
- McHarg, I. L. (1969). **Design with nature**. New York: John Wiley and Sons.
- Microsoft ASP.NET Specification. Disponível em. <<http://www.asp.net/>>
- Microsoft Developer Network Tutorial (2008), MSDN. Disponível em:
<<http://msdn.microsoft.com/vstudio/partners>>
- Parihar M. *et al* (2002). **ASP.NET Bible**, Hungry Minds, Inc., 9-10.
- Pitz, J. C, e Figueiredo, D. (2001). **Trabalho de Introdução à Informática Agrícola**. Sistemas de Informações Geográficas. Disponível em:
<<http://www.inf.ufsc.br/~barreto/trabaluno/ICCPitz20011.pdf>>
- Simon, E., Tomasic, A. **Improving Access to Environment Data using Context Information**, SIGMOD Record; 1997.
- Tschiedel, M. e Ferreira, M. F. (2002). **Introdução à Agricultura de Precisão: Conceitos e Vantagens**. Ciência Rural, Santa Maria, v.32, n.1, p.159-163.
- Tomlinson R. F., (1967). **An Introduction to the Geographic Information System of the Canada Land Inventory**. Canada Department of Forestry and Rural Development, Ottawa.
- Turtschi A. *et al* (2002). **C# .NET Web Developer's Guide**, Syngress Publishing, 1-8.